REMARKS

Claims 1 – 8 are pending. Reconsideration is requested.

Claims 1 – 8 were rejected under 35 U.S.C. 103(a) as being unpatentable over Brustoloni et al. (US. Patent 6,886,103) in view of Sapuntzakis, and further in view of Tsunoda (US. Patent 6,516435).

In prior art methodology, the header information below the TCP was used for processing by the network stack, transparent to the application. In Brustoloni, the AH or IPSec header is below the TCP, since the information in the AH IPSec header is for use by the stack.

Further, in prior art methodology, the data or information in the TCP or above was used by the application. In Sapuntzakis the RDMA information, which is for use by the application, is in the TCP.

Even if Brustoloni had seen Sapuntzakis, Brustoloni would have not had incentive to combine his method with Sapuntzakis because

1) the Brustoloni technology and Sapuntzakis's RDMA are not on the same side of the stack,

2) the Brustoloni technology and Sapuntzakis's RDMA are not in the same layer, and

3) RDMA is used by the application and not by the network stack.

To put the RDMA used by the application below the stack, below the TCP, was against the known methodology. Known methodology did not put information needed by the application, in the header or otherwise, below the TCP. As an example of the standard layer protocol is attached herein in Appendix A and taught in http://en.wikipedia.org/wiki/Internet_protocol_suite. Another picture of the same stack can be found in any standard TCP/IP reference book from 2001, such as "TCP/IP Illustrated, Volume1, The Protocols", by W Richard Stevens, published by Addison-Wesley, 20th printing, November 2001.

Please note that the RDMA information as taught by Sapuntzakis is in layer 4, the Application Layer. The AH IPSec header as taught by Brustoloni is in layer 2, the Internetwork Layer. There is no suggestion or motivation in the generally available knowledge, available either in 2001 or in 2006, that would enable one of ordinary skill in the art to combine the two references.

Furthermore, Tsunoda does not teach moving application information from above the TCP layer to below the TCP layer.

Thus, with due respect, the Examiner has not established a *prima facie* case of obviousness. Applicants respectfully submit that claims 1 -- 8 are allowable over Brustoloni et al. (US. Patent 6,886,103) in view of Sapuntzakis, and further in view of Tsunoda (US. Patent 6,516435).

Applicants believe that the above amendments and remarks are fully responsive to all the objections and grounds of rejections by the examiner. In view of the foregoing amendments and remarks, the applicants respectfully submit that all the pending claims are deemed to be allowable. Their favorable reconsideration and allowance is respectfully requested.

Should the Examiner have any question or comment as to the form, content or entry of this Amendment, the Examiner is requested to contact the undersigned at the telephone number below. Similarly, if there are any further issues yet to be resolved to advance the prosecution of this application to issue, the Examiner is requested to telephone the undersigned counsel.

Please charge any fee associated with this paper to deposit account No. 09-0468.

Respectfully submitted,

By: ____________________

Stephen C. Kaufman
Attorney for Applicant
Registration No. 29,551
Telephone No.: (914) 945-3197

IBM Corporation
Intellectual Property Law Department
P. O. Box 218
Yorktown Heights, New York 10598

Appendix A

Internet protocol suite

http://en.wikipedia.org/wiki/Internet_protocol_suite

From Wikipedia, the free encyclopedia

The **Internet Protocol Suite** is the set of communications protocols that implement the protocol stack on which the Internet and most commercial networks run. It is sometimes called the **TCP/IP** protocol suite, after the two most important protocols in it: the Transmission Control Protocol (TCP) and the Internet Protocol (IP), which were also the first two defined.

The Internet Protocol Suite like many can be viewed as a set of layers, each layer solves a set of problems involving the transmission of data, and provides a well-defined service to the higher layers based on using services from some lower layers. Higher layers are logically closer to the user and deal with more abstract data, relying on lower layers to translate data into forms that can eventually be physically manipulated.

The Internet Protocol Suite can be roughly fitted to the OSI model which describes a fixed set of 7 layers and some vendors like to use this model. However not all of these layers fit well with ip based networking (which really involves a varying number of layers depending on the design of the applications and the underlying network) and some believe that trying to fit the internet protocol suite to the OSI model does more to confuse than to help.

Contents

[hide]

[edit]

Layers in the Internet Protocol stack

There is some discussion about how to map the TCP/IP model onto the OSI model. Since the TCP/IP and OSI protocol suites do not match precisely, there is no one correct answer.

In addition, the OSI model is not really rich enough at the lower layers to capture the true layering; there needs to be an extra layer (the Internetworking layer) between the Transport and Network layers. Protocols specific to a particular network type, but which are run on top of the basic hardware framing, ought to be at the Network layer. Examples of such protocols are ARP and the Spanning Tree Protocol (used to keep redundant bridges idle until they are needed). However, they are local protocols and operate beneath the internetwork functionality. Admittedly, placing both groups (not to mention protocols which are logically part of the internetwork layer, but run on top of the internetwork protocol, such as ICMP) all at the same layer can be confusing, but the OSI model is not complex enough to do a better job.

The following diagram attempts to show where various TCP/IP and other protocols would reside in the original OSI model:

7	Application	e.g. HTTP, SMTP, SNMP, FTP, Telnet, SIP, SSH, NFS, RTSP, XMPP, Whois, ENRP
6	Presentation	e.g. XDR, ASN.1, SMB, AFP, NCP
5	Session	e.g. ASAP, TLS, SSH, ISO 8327 / CCITT X.225, RPC, NetBIOS, ASP, Winsock, BSD sockets
4	Transport	e.g. TCP, UDP, RTP, SCTP, SPX, ATP, IL
3	Network	e.g. IP, ICMP, IGMP, IPX, BGP, OSPF, RIP, IGRP, EIGRP, ARP, RARP, X.25
2	Data Link	e.g. Ethernet, Token ring, HDLC, Frame relay, ISDN, ATM, 802.11 WiFi, FDDI, PPP
1	Physical	e.g. wire, radio, fiber optic, Carrier pigeon

Commonly, the top three layers of the OSI model (Application, Presentation and Session) are considered as a single Application Layer in the TCP/IP suite. Because the TCP/IP suite has a comparatively lightweight session layer, consisting of opening and closing connections under TCP and RTP and providing different port numbers for different applications under TCP and UDP, these functions may be augmented by individual applications (or libraries used by those applications). Similarly, IP is designed around the idea of treating the network below it as a black box so it can be considered as a single layer for the purposes of discussing TCP/IP.

4	Application *(OSI layers 5 through*	e.g. HTTP, FTP, DNS *(routing protocols like BGP and RIP, which for a variety of reasons run over TCP and UDP respectively, may also be*

	7)	*considered part of the Internetwork layer)*
3	Transport *(OSI layers 4 and 5)*	e.g. TCP, UDP, RTP, SCTP *(routing protocols like OSPF, which run over IP, may also be considered part of the Internetwork layer)*
2	Internetwork *(OSI layer 3)*	For TCP/IP this is the Internet Protocol (IP) *(required protocols like ICMP and IGMP run over IP, but may still be considered part of the Internetwork layer; ARP does not run over IP)*
1	Link *(OSI layers 1 and 2)*	e.g. Ethernet, Wi-Fi, MPLS, etc.

[edit]

The link layer

The Link layer is not really part of the Internet protocol suite, but is the method used to pass packets from the Internet layer of one device to the Internet layer of another. This process can be controlled both in the software device driver for the network card, as well as on firmware or specialist chipsets. These will perform data link functions such as adding a packet header to prepare it for transmission, then actually transmit the frame over a physical medium. On the other end, the link layer will receive data frames, strip off the packet headers, and hand the received packets to the Internet layer.

However, the link layer is not always so simple. It may also be a Virtual private network (VPN) or tunnel, where packets from the Internet layer, instead of being sent over a physical interface, are sent using a tunneling protocol and another (or the same) protocol suite. The VPN or tunnel is usually established ahead of time, and has special characteristics that direct transmission out a physical interface does not (for example, it may encrypt the data going over it). This recursive use of the protocol suite can be confusing since the link "layer" is now an entire network. But it is an elegant method for implementing often complex functions. (Though care is needed to prevent a packet that is wrapped and sent through a tunnel being repeatedly re-wrapped and sent down the tunnel again).

[edit]

The Internetwork layer

As originally defined, the Network layer solves the problem of getting packets across a single network. Examples of such protocols are X.25, and the ARPANET's Host/IMP Protocol.

With the advent of the concept of internetworking, additional functionality was added to this layer, namely getting data from the source network to the destination network. This generally involves routing the packet across a network of networks, known as an internet.

In the internet protocol suite, IP performs the basic task of getting packets of data from source to destination. IP can carry data for a number of different higher level protocols; these protocols are each identified by a unique *IP Protocol Number*. ICMP and IGMP are protocols 1 and 2, respectively.

Some of the protocols carried by IP, such as ICMP (used to transmit diagnostic information about IP transmission) and IGMP (used to manage multicast data) are layered on top of IP but perform internetwork layer functions, illustrating an incompatibility between the internet and OSI models. All routing protocols, such as BGP, OSPF, and RIP are also really part of the internetwork layer, although they might seem to belong higher in the stack.

[edit]

The transport layer

The protocols at the Transport layer can solve problems like reliability ("did the data reach the destination?") and ensure that data arrives in the correct order. In the TCP/IP protocol suite, transport protocols also determine which application any given data is intended for.

The dynamic routing protocols which technically fit at this layer in the TCP/IP Protocol Suite (since they run over IP) are generally considered to be part of the Network layer; an example is OSPF (IP protocol number 89).

TCP (IP protocol number 6) is a "reliable", connection-oriented, transport mechanism providing a reliable byte stream, which makes sure data arrives complete, undamaged, and in order. TCP tries to continuously measure how loaded the network is and throttles its sending rate in order to avoid overloading the network. Furthermore, TCP will attempt to deliver all data correctly in the specified sequence. These are its main differences from UDP, and can become disadvantageous in real-time streaming or routing applications with high internetwork layer loss rates.

The newer SCTP is also a "reliable", connection-oriented, transport mechanism. It is record rather than byte oriented, and provides multiple sub-streams multiplexed over a single connection. It also provides multi-homing support, in which a connection end can be represented by multiple IP addresses (representing multiple physical interfaces), such that if one fails the connection is not interrupted. It was developed initially for telephony applications (to transport SS7 over IP), but can also be used for other applications.

UDP (IP protocol number 17) is a connectionless datagram protocol. It is a "best effort" or "unreliable" protocol - not because it is particularly unreliable, but because it does not verify that packets have reached their destination, and gives no guarantee that they will arrive in order. If an Application requires these characteristics, it must provide them itself, or use TCP.

UDP is typically used for applications such as streaming media (audio and video, etc) where on-time arrival is more important than reliability, or for simple query/response applications like DNS lookups, where the overhead of setting up a reliable connection is disproportionately large.

DCCP is currently under development by IETF. It provides TCP's flow control semantics, while keeping UDP's datagram service model visible to the user.

Both TCP and UDP are used to carry a number of higher-level applications. The applications at any given network address are distinguished by their TCP or UDP *port number*. By convention certain *well known ports* are associated with specific applications.

RTP is a datagram protocol that is designed for real-time data such as streaming audio and video. RTP is a session layer that uses the UDP packet format as a basis yet is said to sit within the transport layer of the Internet protocol stack.

[edit]

The application layer

The Application layer is the layer that most common network-aware programs use in order to communicate across a network with other programs. Processes that occur in this layer are application specific; data is passed from the network-aware program, in the format used internally by this application, and is encoded into a standard protocol.

Some specific programs are considered to run in this layer. They provide services that directly support user applications. These programs and their corresponding protocols include HTTP (The World Wide Web), FTP (File transport), SMTP (Email), SSH (Secure remote login), DNS (Name <-> IP Address lookups) and many others.

Once the data from an application has been encoded into a standard application layer protocol it will be passed down to the next layer of the IP stack.

At the Transport Layer, applications will most commonly make use of TCP or UDP, and server applications are often associated with a well-known port number. Ports for server applications are officially allocated by the Internet Assigned Numbers Authority (IANA) but developers of new protocols today often choose the port numbers themselves. As it is rare to have more than a few server applications on the same system, problems with port conflicts are rare. Application software also generally allows users to specify arbitrary port numbers as runtime parameters.

Client applications connecting out generally use a random port number assigned by the operating system. Applications that listen on a port and then send that port to another copy of the application via a server to set up a peer-peer link (e.g. dcc file transfers on IRC). May also use a random port but the applications usually allow specification of a specific port range to allow the ports to be mapped inwards through a router that implements network address translation.